УДК 004.75

АРХІТЕКТУРА АНАЛІТИЧНОЇ СИСТЕМИ ДЛЯ ВИЯВЛЕННЯ ШАХРАЙСЬКИХ ТРАНЗАКЦІЙ

Ю. А. Лісік[1], М. В. Грайворонський[1]

[1] *Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Фізико-технічний інститут*

Анотація

У цій роботі описана розробка і реалізація розподіленої, високомасштабованої, відмовостійкої, аналітичної системи для виявлення шахрайських транзакцій. Система працює в режимі реального часу і використовує визначені алгоритми машинного навчання для здійснення прогнозувань результату транзакцій. Мета роботи полягає в створенні доступної ефективної системи, яка максимально допоможе сторонам-учасникам (мерчантам, платіжним системам, банкам) виявити шахрайські транзакції в режимі реального часу. Знайдено і описано ряд архітектурних і програмних підходів, які призвели до скорочення витрат на апаратні засоби, програмне забезпечення і персонал, для обслуговування системи.

Ключові слова: шахрайство з кредитними картами, машинне навчання, інтелектуальний аналіз даних, хмарні технології.

Вступ

Щодня в світі здійснюються тисячі хакерських атак, які кожного разу стають все винахідливішими та досконалішими. Шахраї атакують не лише банкомати та термінали, але і віртуальні рахунки.

За даними Національного банку України, з кожним роком кількість шахрайських операцій продовжує зростати. Як свідчить статистика, майже в половині українських банків було зафіксовано факти шахрайства з картками. Навіть якщо шахрайство займає менше відсотка від загального обсягу карткових операцій в Україні, воно продовжує розвиватися і третина скарг, які надходять до Нацбанку від громадян – це саме заяви на дії злочинців.

За рік до НБУ надійшло звернення від 47 банків (або 48 % від кількості всіх банків-членів платіжних систем), які заявили про факт шахрайства з платіжними інструментами. Головним завданням усіх міжнародних платіжних систем є виявлення і перешкоджання шахрайським операціям. Безпеку онлайн-платіжних транзакцій відстежують безліч систем на різних рівнях і етапах проходження платежу.

Швидке зростання кількості операцій з пластиковими картами, що проводяться через інтернет, ставить перед розробниками систем прийому онлайн-платежів все нові й нові виклики, пов'язані з ростом масштабу таких систем та ускладненням підходів до забезпечення їх надійності та безпеки.

Мерчант (веб-застосунок, який приймає оплату) часто стикається з таким родом проблем як шахрайські транзакції і його власники повиннні намагатися якось на них реагувати, оскільки в кращому випадку, коли відбудеться шахрайство, то вони не отримають гроші, а в гіршому – їх оштрафують.

Так як власники мерчанту не можуть виділити потрібних коштів на захист, оптимальним варіантом зниження витрат може стати введення додаткових ускладнених перевірок для клієнтів і делегування частини обов'язків. Одним із таких способів може бути *3D Secure* (делегування обов'язків перевірки банку-емітенту).

3D Secure – це протокол, який являє собою двохфакторну аутентифікацію власника кредитної чи дебетової карток що використовується при онлайн-покупці. Але такий вид захисту, що вимагає від користувача додатково виконувати завдання, може призвести до різкого зменшення кількості успішно завершених транзакцій.

Більшість існуючих робіт описують великий і постійно зростаючий список методів інтелектуального аналізу даних у виявленні шахрайства з кредитними картами. Вони включають в себе нейронні мережі [1, 2, 3], метод опорних векторів (SVM)[4] і самоорганізованих карт (SOM)[5]. Класифікація, як і раніше, безумовно,найбільш поширений тип інтелектуального аналізу даних завдань в шахрайстві з кредитними картами[6]. [7] є однією з найбільш ранніх робіт по розподіленому аналізу даних при виявленні шахрайських транзакцій. Автори звернули увагу на масштабовані методи аналізу великих обсягів даних і визначення ефективних детекторів шахрайства у необхідний проміжок часу. Але ніде не було описано архітектуру системи для виявлення шахрайських транзакцій.

Тому було проведено експеримент, метою якого було створити розподілену високомасштабовану відмовостійку систему виявлення шахрайських платежів.

1. Вимоги до системи

Загальна схема роботи практично будь-якого механізму фрод – моніторингу виглядає наступним чином: в момент здійснення оплати за допомогою банківської карти збирається кілька показників – починаючи від IP-адреси комп'ютера і закінчуючи статистикою оплат по цій карті. Система має набір евристичних алгоритмів, тобто лімітів фільтрів безпеки. Кожен з фільтрів перевіряє користувача – його персональні і карткові дані. Мета системи – переконатися в тому, що користувач є реальним власником карти, за допомогою якої здійснюють покупку на сайті. У разі виявлення підозрілої активності, тобто перевищення будь-якого значення параметру, фільтр автоматично блокує можливість здійснення платежу за цією карткою.

При вводі платіжних даних, мерчант не завжди перевіряє коректність їх написання. Аби заощадити ресурси CPU і запобігти зашумленню навчальної моделі, достатньо просто виявити помилки в платіжних реквізитах, оскільки це може бути як помилка користувача, так і зловмисні дії.

Тому потрібно перевіряти чи містить ім'я власника хоча б дві літери, чи дійсна на сьогоднішній час платіжна карта, а також чи проходить платіжна карта перевірку алгоритмом Луна.

Даний алгоритм використовується для розрахунку контрольної цифри номера пластикової карти. Він призначається для того аби виявити помилки, які користувач міг ненавмисно допустити при вводі номера картки і може з деякою мірою вірогідності говорити про відсутність помилок в номері карти.

Для виявлення шахрайських транзакцій існує велика кількість евристик. Деякі антифрод системи використовують до 150 евристик. Але, це не завжди ефективно. Велика кількість евристик дає лише перенавчену модель, неправильне розпізнавання, зменшення продуктивності програми та велике завантаження системи.

Нижче описано основні найефективніші евристичні алгоритми, які будуть використовуватись у спроектованій системі для того, щоб зменшити кількість шахрайських транзакцій і, відповідно, повернення коштів банку-емітенту (*chargeback*):

- клієнт змінив особисту інформацію (включаючи пароль) перед здійсненням високовартістної транзакції
- декілька неуспішних спроб авторизації перед транзакцією
- ім'я власника картки не збігається з іменем клієнта на сайті
- країна клієнта карти не збігається з країною власника аккаунта
- оплата проходить у нічний час (по локальному часу клієнта)
- один клієнт використовує багато карток (особливо різних банків)

Часто головним підходом є присвоєння фіксованого значення для якогось із фільтрів. Але, в такому підході дуже багато недоліків. Наприклад, буде багато помилкових відхилень легітимних платежів та пропуск фроду при невеликій зміні стратегії шахраями. Тому найкращим рішенням даної проблеми буде розробити систему, в якій евристичні фільтри здатні до самонавчання як на накопиченій історії транзакцій користувача, так і відповідно до його пристрою і поведінки користувача на сайті.

Також необхідно створити глобальні чорні списки для блокування транзакцій. До таких списків можна віднести чорний список банківських карт, IP-адресів, країн, мерчантів і т.д.

Окрім, перевірки стандартних параметрів пристрою користувача, таких як IP-адреса, країна, тип браузеру та ін., у системі використовуються новітні технології для аналізу, наприклад, *«Font Fingerprint»* – (технологія отримує список всіх шрифтів на системі користувача і з них робить унікальний «зліпок» системи), *«Mouse Fingerprint»* – («зліпок» руху мишки користувача), *«AudioContex»* (технологія зняття відбитка аудіо на пристрої за допомогою JavaScript), *«WebGL»* (можна дізнатись виробника відеокарти пристрою користувача і таким чином визначити чи використовується віртуальна машина, чи ні) та інші.

Проектована система для виявлення шахрайський транзакцій є критично важливою бізнес-системою, тому її зупинка може призвести до зупинки прийому оплати або до збільшення фінансових і репутаційний ризиків компанії (якщо система працює некоректно). Тому система має бути, розподіленою, відмовостійкою, високомасштабованою і надійною.

Згідно міжнародного стандарту сертифікації *PCI DSS* дозволяється зберігати перші 6 і останні 4 цифри карти і не заборонено проводити генерування внутрішніх ідентифікаторів для клієнтських карток і передавати по захищених каналах імена власників та термін закінчення карток. Згідно положень закону України «Про захист персональних даних», всі програмні модулі будуть працювати з тільки деперсоніфікованими даними.

2. Архітектура системи

Систему складають кілька компонентів, що використовують хмарну платформу *Microsoft Azure*(рис. 1):

1) *Service Machine Learning* – основний сервіс, що виявляє шахрайські операції. Він заснований на алгоритмах машинного навчання.
2) *Transactions Database* – база, в якій зберігається уся інформація.
3) *API Service* – сервіс надання API з метою забезпечення інтеграції з *Service Machine Learning.*

Інфраструктура зумовлює наявність значної кількості обмежень, які необхідно врахувати на рівні архітектури. Те, що стосується обмежень юридичного характеру, вже обговорювалось, а пов'язані з вибором платформи *Microsoft Azure* переваги та обмеження будуть висвітлюватись далі. Для роботи знадобляться різні сервіси *Azure*, зокрема, *Machine Learning*, черги, *Table* та *Cloud service* для ролей.

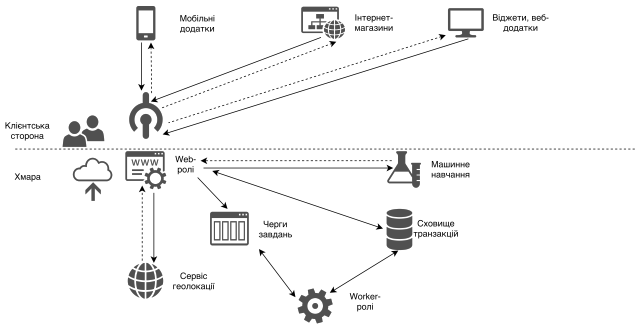

Рис. 1. Архітектура системи

Інфраструктура не потребуватиме майже жодних грошових витрат.

Крім того, від використання згаданих сервісів можна отримати такі переваги:

- безпечне зберігання (згідно стандартів інформаційної безпеки ISO/IEC)
- надійне зберігання даних завдяки шардингу і реплікації
- з урахуванням навантаження може автоматично здійснюватись масштабування кількості робочих машин. На базі *PartitionKey* розподіляються таблиці *NoSQL*-сховища
- стійкість до відмов. Не тільки можна, а й рекомендовано запуск робочих вузлів у кількох примірниках
- високий ступіть доступності

Можливість скористатись усіма згаданими перевагами вдається отримати виключно завдяки застосуванню таких архітектурних підходів:

- відбувається виключно асинхронна взаємодія через мережу. При цьому обов'язково застосовуються retry-політики
- структурована або напівструктурована інформація зберігається з використанням горизонтального шардингу
- використовується протокол передачі даних, що не передбачає збереження стану
- повідомлення збираються у черги, завдяки чому навантаження вирівнюється, а завдання гарантовано опрацьовуються
- до даних про транзакції додається поле *timestamp*, що дозволяє убезпечити лог транзакцій від блокувань

Слід враховувати наявність певних обмежень у хмарних сервісах, зокрема, розмір повідомлення чи граничну чисельність запитів, що надсилаються одночасно.

Що стосується мерчанту, то система виступає в ролі сервісу, з яким через *API Service* можна взаємодіяти по https-протоколу. *API Service* функціонує у складеному з кількох веб-ролей, тобто шарів програми, кластері.

Послідовність дій:

1) направлення запиту з даними про транзакцію
2) трансформування моделі (за схемою MVC)
3) спрямування запиту до системи, що передбачає результат транзакції
4) отримання результату. Надходить відповідь, чи буде успішним платіж
5) збереження інформації
6) надання даних клієнту
7) перенавчання моделі. Навчальна вибірка оновлюється та перераховується
8) (опційно) за ініціативою клієнта у випадку, якщо спостерігаються розбіжності між фактичним результатом транзакції та передбаченням, надсилається запит з даними про результат платежу.

З метою підвищити якість функціонування алгоритму передбачення клієнтам надається набір інструментів, за допомогою яких можна уточнити результати платежу. Наприклад, у випадку наявності розбіжностей між даними, наданими системою, та фактичним результатом існує можливість надіслати запити з метою уточнення інформації. Формат таких запитів – *IdTrans*, *ResultTrans*, *LastUpdateTime*. Вони проходять обробку *API Service* та після здійснення валідації переміщуються у стійкий до відмов сервіс черг *«Queue»*. Один з роботів по одному вилучає запити з черги.

Збереження інформації про операції та додаткових даних, що з ними пов'язані, відбувається у довгостроковому сховищі транзакцій на базі стійкого до відмов *NoSQL*-сховища *Table*. Лог транзакцій – це 2 таблиці. 1 – Зі статистичними метриками *StatTrans*. У цій таблиці міститься інформація про кількість платежів з певної картки, успішних операцій, а також IP-адрес, про часові інтервали між транзакціями, час реєстрації покупця на веб-ресурсі тощо 2 – З фактами, що стосуються транзакції, *InfroTrans*. Зазначається номер операції за порядком, номер мерчанта, хеш імені власника картки, якщо воно було вказано, сума транзакції тощо. На 7 – 8 кроках відбувається перенавчання моделі.

Останні дані по транзакціях містяться у сховищі відповідного журналу, тому інформація з нього виступає в ролі навчальної вибірки. Для перенавчання може передбачатись розклад, сигналом для його проведення може бути виникнення певного значення нових записів у журналі або перевищення певної межі помилкових прогнозів.

Зупинимось докладніше на алгоритмі машинного навчання. Про головні припущення та концепції, що мають певну користь в процесі створення моделі, йшлось вище. Розробка якісної гіпотези являє собою ітеративну процедуру, що не обходиться без проб та помилок. Базою для неї є знання у відповідній галузі, що досліджується, а також в сфері машинного навчання.

Пакет даних для моделі виявлення шахрайських транзакцій – це відповідний лог, що складається з двох таблиць. Одна з них містить заздалегідь розраховані статистичні метрики, а друга – факти про операції.

На стадії отримання інформації завантажуються згадані таблиці за допомогою елемента керування *Reader*. Він дає змогу завантажувати структуровані та напівструктуровані пакети даних. При цьому, джерелом можуть бути реляційні та нереляційні бази

даних (наприклад, *NoSQL*). Є можливість завантаження різних форматів текстових документів. Крім того, інформація може вноситись вручну за допомогою інструменту *Enter Data*.

Непоодинокі випадки, коли пакет містить інформацію, що дублюється, а це призводить до зниження точності передбачення. Для вилучення зайвих даних використовується інструмент *Remove Duplicate Rows*. Така інформація, як IP-адреса та країна не завжди може бути визначена, тому відповідні поля залишаються порожніми. Використовуючи інструмент *Clean Missing Data*, в пустих полях, де повинні бути вказані назви країн, зазначимо *«empty»* та видалимо зайві рядки, в яких відсутня інформація про валюту або суму операції.

В комплект поставки *Azure ML Studio* включений модуль нормалізації даних, завдяки якому значення пакету приводиться у відповідність із загальною шкалою шляхом застосування математичної функції числових даних. За загальним правилом, застосування модуля передбачається для всіх колонок з числами, однак є можливість виділити окремі стовпчики. Можна також обрати математичну функцію, наприклад, *Min-Max* або *Z-Score*.

Інструмент *Normalize Data* дозволяє здійснити *Z-Score*-нормалізацію інформації, наприклад, про суму транзакції або інші великі числа. Пакет даних, що отриманий за результатами проведеної роботи, ділимо на тестову та навчальну вибірки. Далі слід обрати найкраще співвідношення даних з двох вибірок. Інструмент *Split* дозволяє розбити набір даних. Реалізуються різні стратегії поділу та зазначаються пропорції інформації в кожному з піднаборів. За допомогою згаданого інструменту ми відділимо 70 відсотків даних та передамо їх до навчальної вибірки, одночасно ввімкнувши під час розподілу змішування в довільному порядку, використавши для цього функцію *Randomized split*. Завдяки змішуванню інформації в процесі розподілу даних не відбувається перевантаження, яке може бути зумовлене значними витоками карткових номерів.

Сформуємо декілька алгоритмів класифікації та оберемо на тестовій вибірці той з них, який дає найбільш точні дані. Варто зауважити, що ціна шахрайських операцій, які не були виявлені (*False Negative*), суттєво перевищує ціну транзакцій, що помилково сприйняті, як шахрайські (*False Positive*). На фактичних даних зовсім не обов'язково вдасться отримати таку ж продуктивність, як на тестових. Це зумовлює необхідність проаналізувати, які упущення спостерігаються в моделі, виявити причини, що зумовлюють набагато кращий чи гірший результат по певних алгоритмах. Далі слід внести відповідні корективи та знов запустити алгоритм навчання. Такі дії мають повторюватись до тих пір, поки дослідник матиме модель, що прийнятна за показником точності. Ми використаємо у своєму дослідженні декілька таких алгоритмів двокласової класифікації, як:

- нейромережа;
- метод опорних векторів;
- дерево рішень (*Boosted Decision Tree*);
- логістична регресія.

Висновки

Представлено архітектуру системи виявлення шахрайських транзакцій. Використовуючи машинне навчання, запропонована система скорочує початкові витрати на інфраструктуру та програмне забезпечення практично до нуля. Було проведено тестування 4 моделей і в цілому найкращу точність показав алгоритм двокласової нейронної мережі – *Two-Class Neural Network*, за ним алгоритм на основі дерев рішень – *Two-Class Boosted Decision Tree*. В майбутньому планується розширити евристики і максимально точно навчити модель на реальних даних, тому що машине навчання вимагає дуже тонкого і детального налаштування параметрів, щоб отримати дійсно оптимальну продуктивність класифікаторів.

Перелік використаних джерел

1. Aleskerov E., Freisleben B., Rao B. CARDWATCH: a neural network based database mining system for credit card fraud detection. — 1997. — Access mode: `http://dx.doi.org/10.1109/CIFER.1997.618940`.
2. Brause R., Langsdorf T., Hepp M. Neural data mining for credit card fraud detection. — 1999. — Access mode: `http://ieeexplore.ieee.org/document/809773/`.
3. Neural fraud detection in credit card operations. / J. R. Dorronsoro, F. Ginel, C. Sánchez, C. Santa Cruz. — 1999. — Access mode: `http://dx.doi.org/10.1109/72.595879`.
4. Lu Q., Ju C. Research on credit card fraud detection model based on class weighted support vector machine. — 2011. — Access mode: `http://dx.doi.org/10.4156/jcit.vol6.issue1.8`.
5. Bansal M., Suman G. Credit Card Fraud Detection Using Self Organised Map // International Journal of Information and Computation Technology. — 2014.
6. The application of data mining techniques in financial fraud detection / E. W. T. Ngai, Y. Hu, Y.H. Wong et al.
7. Distributed data mining in credit card fraud detection. / P.K. Chan, W. Fan, A.L. Prodromidis, S.J. Stolfo. — 1999. — Access mode: `http://dx.doi.org/10.1109/5254.809570`.